Generation and Evaluation of Business Continuity Processes

Using Algebraic Graph Transformation and the mCRL2 Process Algebra

Christoph Brandt[1], Frank Hermann[2], Jan Friso Groote[3]

[1] Université du Luxembourg, SECAN-Lab, Campus Kirchberg,
6, rue Richard Coudenhove-Kalergi, 1359 Luxembourg-Kirchberg, EU
e-mail: `christoph.brandt@uni.lu`,
WWW home page: `http://wiki.uni.lu/secan-lab`

[2] Technische Universität Berlin, Fakultät IV,
Theoretische Informatik/Formale Spezifikation, Sekr. FR 6-1,
Franklinstr. 28/29, 10587 Berlin, EU
e-mail: `frank@cs.tu-berlin.de`,
WWW home page: `http://www.tfs.tu-berlin.de`

[3] Eindhoven University of Technology, Systems Engineering,
Department of Computer Science. Hoofdgebouw kamer 6.75,
Den Dolech 1 P.O. Box 513, 5600 MB Eindhoven The Netherlands, EU
e-mail: `J.F.Groote@tue.nl`,
WWW home page: `http://www.win.tue.nl/~jfg`

Critical business processes can fail. Therefore, continuity processes are needed as back-up solutions. Today, those continuity processes are set up and maintained manually. They are mostly based on best-practices that focus on specific continuity scenarios. Nevertheless, failures can occur in new and unforeseen combinations. As a consequence, a given business continuity plan needs to handle such situations as well. For this purpose, we present a technique for the generation and validation of the universe of continuity processes given a critical business process at Credit Suisse. The presented approach uses a combination of formal methods in the area of algebraic graph transformation and process algebra encompassing modal logic. The overall approach prepares for a sound evaluation of the effectiveness and efficiency of such plans. It uses formal tools, not standard software engineering solutions, to benefit from formal guarantees that facilitate the implementation of local and global security requirements.

ACM Classification: C.3 Special-Purpose and Application-based Systems (Process control systems), G.2.3 Applications, I.6.4 Model Validation and Analysis, I.6.5 Model Development

Keywords and Phrases: business continuity, business process, algebraic graph transformation, process algebra, generation, evaluation, enterprise modeling

1 Introduction

Given a business process model in a domain specific modeling language together with security and continuity rules, the problem addressed in this paper is how to set up and determine a continuity plan in the organizational context at Credit Suisse, which can be evaluated regarding its effectiveness and efficiency. The research question is how to give an answer to continuity issues of the bank that is sound and compatible with a decentralized organization which is a hard organizational side-constraint. The scope of this study is to show how to model a simplified version of a real-world business process and continuity situations compatible with these real-world requirements in order to derive all possible process variants and continuity processes that match security rules. It prepares for the evaluation of the effectiveness and efficiency of a continuity plan that is left for future work.

The novelty presented consists of a combination and conceptual integration of fully implemented formal methods – in contrast to standard software engineering solutions – that can be used in a transparent way to address business continuity issues regarding critical business processes at Credit Suisse by using declarative formal techniques. This paper extends the concepts and results presented in (Brandt, Hermann, and Engel 2009a) concerning the analysis of business processes and the generation of validated process variants for business continuity. The first main contribution consists of applying the formal technique of algebraic graph transformation to generate the universe of continuity processes given a critical business process and its continuity snippets. In addition to what have already shown in (Brandt et al. 2009a), we present here how the generation of process variants is executed based on a derived formal process model (Corradini, Hermann, and Sobociński 2008). The graph technique helps to ensure *local* security requirements by declarative rules. The second main contribution that has not been shown in (Brandt et al. 2009a) consists of demonstrating how the formal technique of algebraic graph transformation is used to realize a model transformation by the help of triple graph grammars to derive an *mCRL2* model representing a possible denotational semantics for the given EPC. This *mCRL2* model is further used to check for *global* security properties of the business process.

This paper is organized as follows: Firstly, the notion of business continuity is introduced as it is discussed in the literature and practiced at Credit Suisse. Secondly, modeling business continuity processes is put in the context of a modeling framework. Thirdly, a simplified real-world loan granting process is introduced as a typical example of a critical business process. It is presented as a human-centric business process model. Here, it is demonstrated how a variant of an event driven process chain together with a set of continuity snippets can serve as a means to derive all possible process variants as well as corresponding continuity processes that respect given local security side-constraints. Fourthly, as a formal technique to generate all possible continuity processes a reconstructed graph grammar is introduced. Fifthly, it is shown how model transformation based on algebraic graph transformation can be used to create an *mCRL2* model as a machine-centric business process model that is going to be used to check for its *global* data-flow and information-flow properties of the business process at stage. Finally, some selected pointers of related work are discussed based on relevant requirements that came out of the banking scenario. Because of the size of the models the specification of the graph grammars as well as the *mCRL2* model encompassing its evaluation are presented in the online version of the corresponding technical report (Brandt, Hermann, and Groote 2010a).

2 Business Continuity

Business Continuity (BC) is introduced here according to the British Standard (BS) 25999-1:2006 and the BS 25999-2:2007 (Zalewski, Sztandera, Ludzia, and Zalewski 2008, Boehmer 2009) by taking two different perspectives: the first one is based on the current scientific literature and industrial practice, the second one is based on concrete requirements of Credit Suisse that came up during this study.

From the point of view of scientific literature and industrial practice BC addresses questions of how to handle risk issues in the case that critical business processes of an organization fail (Cassel 2005). Historically, BC addressed IT processes, later on, business processes came up as the final purpose of their supporting IT processes (Zambon, Bolzoni, Etalle, and Salvato 2007, Asnar and Giorgini 2008). BC addresses the risk that a company may go out of business because a process is not up and running. Creating a continuity plan is a long-term process and companies should review the existing documentation as an ongoing project (Kepenach 2007, Lam 2002). However, continuity plans are quite often just textual documents, which limit possible analysis techniques,

such that a full assessment would require the real execution of a plan or its parts. In the case that modeling techniques are used they often do not match the underlying organizational requirements and they are not sufficiently formalized (Zalewski et al. 2008). Regarding IT, service continuity management (SCM) forms one of the management modules in daily IT service management in the IT Infrastructure Library Service Delivery (Stych and Zeppenfeld 2008, Wan 2008). One important quality of SCM is that failures can be fixed within the Maximum Tolerable Period of Disruption (MTPD) (Zambon et al. 2007). BC is not only relevant for large organizations but also for mid-sized ones as caused due to recent changes in legislation such as HIPPA (Landry and Koger 2006a). The Chartered Management Institute indicated that loss of IT capacity and loss of telecommunications are the main incidents that cause a business disruption (Helms, Van Oorschot, Herweijer, and Plas 2006).

From the point of view of Credit Suisse BC is just one organizational task that has to be integrated with others as there is a need to implement and manage security, risk and compliance requirements in the broader context of enterprise modeling (Brandt, Hermann, and Engel 2009c). All these activities require organizational models like the description of services, processes and rules. However, by today, there is little reuse of such models between the groups responsible for the different areas. As a consequence, results have a low maturity, they are neither soundly integrated nor synchronized and they are not up-to-date with the evolving organization as well as organizational regulations. This situation leads to severe quality problems and high costs. For example, because of the lack of integration, there is no guarantee that business continuity processes respect the security requirements (Quirchmayr 2004). So, a central motivation of Credit Suisse is to look at BC from a holistic organizational perspective (Asnar and Giorgini 2008) and not, as by today, from a bouquet of non-integrated codes of practices derived from external regulations as e.g. Basel II (Basel Committee on Banking Supervision 2006) or the Sarbanes-Oxley Act (United States Code 2002). We will address this issue by providing a modeling framework for an organization like the one of Credit Suisse that can integrate BC as one technique among others in a compatible way. The second motivation of Credit Suisse is to enforce automation to reduce daily costs. We will address this by looking at implementable formal methods that can be used in a declarative way. In detail, Credit Suisse expects processes to be documented by the help of some event-driven process chain (EPC), as discussed in (Zalewski et al. 2008), and that back-up solutions for business continuity can be provided by local process fragments – we like to call them snippets – that can be provided by different people compatible with the decentralized nature of the organization. Based on the process description and the available continuity snippets the universe of continuity processes should be created automatically (van der Aalst, Dumas, Gottschalk, ter Hofstede, Rosa, and Mendling 2008). Finally, Credit Suisse likes to evaluate the effectiveness and efficiency of continuity plans in a methodological and tool supported way. Therefore, it is vital to implant proactive, self-managed systems that ensure continuous operations and, therefore, business continuity (Zambon et al. 2007).

3 Enterprise Modeling

In the context of this study we will build our solution on top of a framework for small, independent and decentralized as well as lean organizational models. We apply the formal technique of algebraic graph transformation to build, change, integrate, transform and synchronize these models using their underlying abstract syntax graphs. In this way the technique covers a wide range of domain specific languages. Graph transformation provides a declarative approach and the execution as well as analysis can be automated. We propose to realize this by the help of a graph transformation engine as a new type of base-system like there are databases or workflow engines available today (Braatz and Brandt 2010). This allows us to provide a framework (Brandt

et al. 2009c) for model integration, transformation and synchronization, which additionally – as a by-product – offers the generation of the universe of business continuity processes based on a given business process together with its continuity snippets. We believe that, by doing this, a maximum of functional reuse of a graph transformation engine will be possible, such that the overall costs of a holistic solution are reduced. We further believe that the use of the same technique for different purposes substantially improves the integration of different methodologies in comparison to today's highly focussed products that handle security, risk and compliance issues of enterprise models implementing best-practices non-homogeneously.

In detail, the new framework for enterprise modeling as shown in Fig. 1 tries to be as close as possible at people's perceptions of organizational structures in the field. From the point of view of conceptual domain languages, IT experts at Credit Suisse like to think in service models, whereas business experts are primarily focussed on processes. In addition to that, compliance groups prefer rule sets. Furthermore, the organizational knowledge is distributed in the organization and for this reason, we propose that experts can directly provide their codified knowledge as fragments of organizational service, process and rule models. The formal techniques of algebraic graph transformation which we assume to be implemented as a graph transformation engine will support the integration of these fragments organized by the modeling framework towards one holistic view in an automated or semi-automated way.
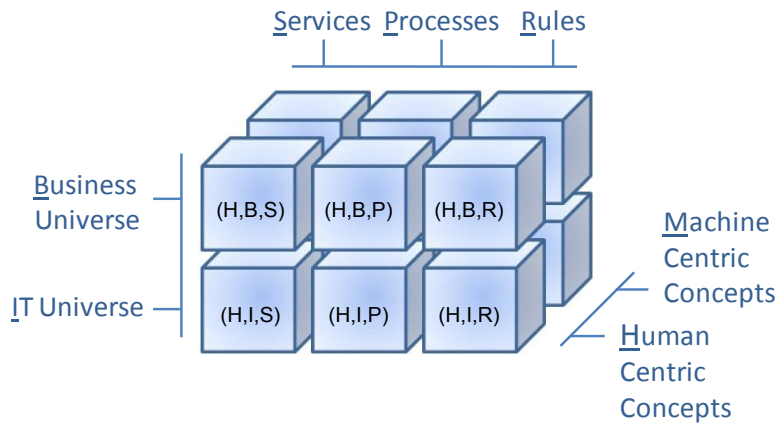
Fig. 1 Holistic Enterprise Modeling Framework for Services, Processes and Rules

However, conceptual domain languages are often not fully formalized and they are by their deep nature inherently evolving over time. For this reason they are not suitable for fully automated evaluation techniques like model checking, despite the fact that they offer a high degree of usability. In contrast to that fully implemented formal methods like first-order logic or process algebra are very suitable for such techniques. However, people in the field do not feel comfortable to work with them. Therefore, we believe that the dominant strategy will be to get the best out of both worlds by combining them and claim that it is necessary to use both, on the one hand human-centric organizational models and on the other hand machine-centric organizational models. They are formally integrated, synchronized and transformed using their abstract syntax by the help of algebraic graph transformation techniques. Requirements of the real-world scenario at Credit Suisse show that it is promising to work with a set of flexible integration rules that are best suited to realize a loose coupling between human-centric and machine-centric models, so that domain languages can still evolve even when implemented formal methods do not change.

In order to support a sound alignment between organizational models of IT and business departments the same types of models – namely organizational service, process and rule models – are used for both universes. This directly shows the need for IT experts at Credit Suisse to model their processes as well as for business experts to codify their business services. It further makes

clear that compliance groups need to focus their rule sets properly towards either business or IT models, which is not the case today. The practice at Credit Suisse shows that there is no clear top-down or bottom-up relationship between business and IT models but a relationship between equals, which contradicts assumptions of enterprise modeling techniques used at Credit Suisse.

The scope of this paper within this modeling framework is to show how a business process given by a human-centric business process model can be used as the input for algebraic graph techniques in order to generate all possible variants respecting the inherently defined data-flow dependencies. Once, continuity snippets are given that show how to back up certain failures in the given business process, the set of continuity processes can be generated using these techniques. Finally, given the human-centric business process models we will sketch how to apply model transformation techniques to create machine-centric process models based on process algebra with the overall purpose in mind to check for data-flow and information-flow qualities the graph techniques cannot test because they only check local properties of models. The following figure (Fig. 2) shows which section of the model framework we are going to address in this paper.
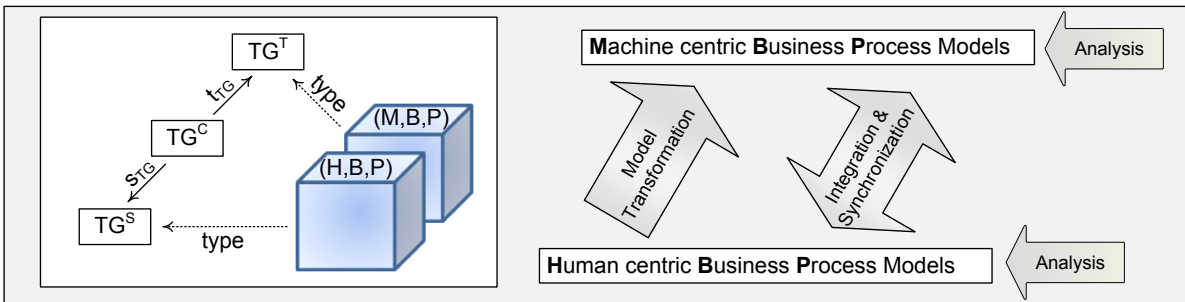

Fig. 2 Scope of the Paper

As visualized in Fig. 2, we consider machine-centric (M, B, P) and human centric business process models (H, B, P). The applied techniques are based on the underlying abstract syntax graph of each model. In order to perform a model transformation from human centric business models (EPCs) to machine-centric business process models ($mCRL2$) we distinguish between the type graph TG^S for the source language and TG^T for the target language. Model transformation as well as integration and synchronization are performed using the triple graph transformation approach (Schürr 1994) as presented in (Brandt et al. 2009a, Ehrig, Ermel, Hermann, and Prange 2009). This way the type graph TG^S is connected with the type graph TG^T via a correspondence type graph TG^C and the connecting graph morphisms $s_{TG} : TG^C \rightarrow TG^S$ and $t_{TG} : TG^C \rightarrow TG^T$ as shown in Fig. 2. The concrete correspondences depending on certain substructures are defined by triple rules in the form of intuitive patterns showing how an already integrated model would evolve.

4 A Loan Granting Process

In the context of this paper we will use a simplified loan granting process as it is used by Credit Suisse in the daily business. The process model is shown in Fig. 3 as a human-centric business process model given by a variant of an event-driven process chain (EPC) (Scheer 2001). We use EPCs because Credit Suisse requested to use existing business process models that already have been created in the past for different purposes. From the point of view of Credit Suisse this simplified loan granting process is critical and several failures can occur. Here, we focus on the availability of organizational agents and IT applications. We will show how additional process fragments, called continuity snippets, can be used to generate and provide back-up solutions, where parts of the
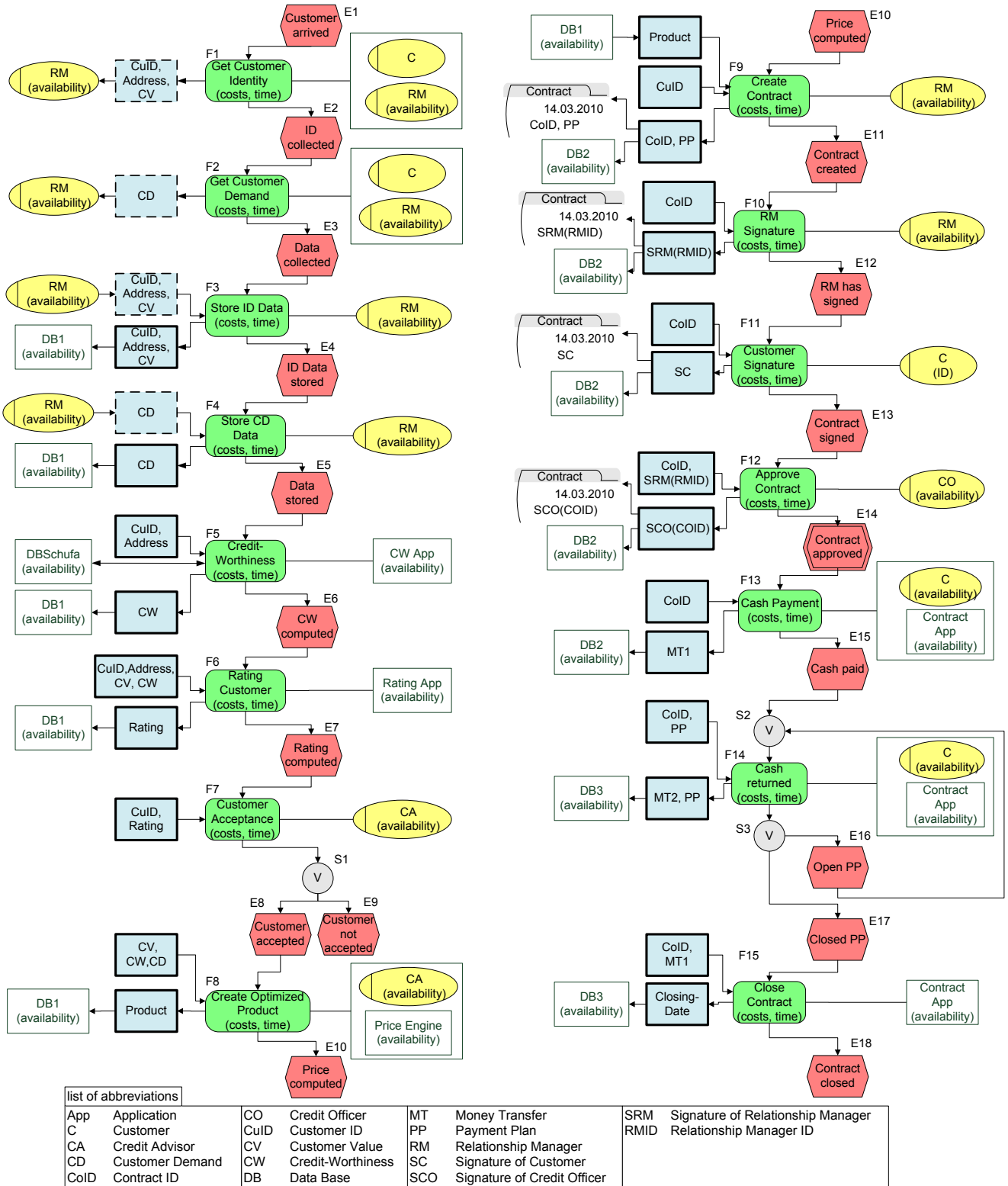
Fig. 3 Workflow of LG

standard process are replaced by the snippets. The snippets can be kept small and simple. They are automatically combined by the graph technique to more sophisticated constructions later on. Based on this process model Sect. 5 presents how the universe of process variants handling different combinations of failures can be generated, where also the equivalent reorderings of the steps are computed. We show how declarative functional requirements and security requirements are ensured during the generation – something, which is currently not manageable by hand, and which is therefore missing in today's practice at Credit Suisse.

The loan granting process is presented by an extended EPC. We use organizational entities like people and applications once at the right side as agents that execute certain business functions and we use them at the same time at the left side as possible sources and sinks of data-flows going into and coming out of business functions. This way, data-flow dependencies become explicit and process variants can be generated based on these dependencies. Furthermore, a buffer for flowing data is made explicit and visualized by boxes between these elements. We assume that such a buffer is managed by a workflow engine. To make clear which parts of the process are executed by such an engine and which are executed by hand, boxes have either a thick or a dashed line. This allows us to cover automated and non-automated parts of a business process by one single human-centric business process model as it was requested by Credit Suisse. The data which is cached by a workflow engine can be reused if needed. We will call this extended EPC a workflow engine based and data-flow oriented EPC, in short: WDEPC (Brandt et al. 2009a).
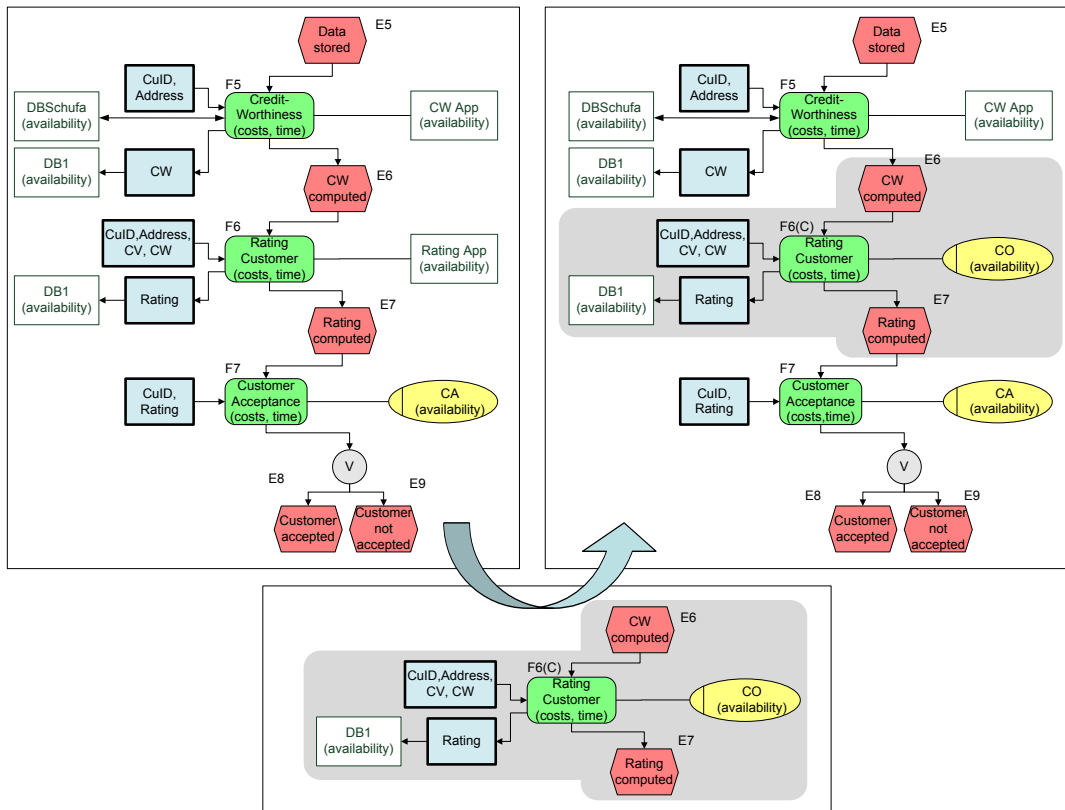
Fig. 4 Alternative Continuity Snippet "Rating Customer" (1 of 16 simple snippets) and Part of the Alternative Workflow

While the given process model shows a fixed order of certain functions, many of them are not directly dependent on each other. But they are partly dependent on the data that is produced by previous steps. Look e.g. at functions "F1" to "F4" for getting and storing certain customer data. Clearly, "F1" needs to be executed before "F3" and similarly "F2" before "F4", but there are no further dependencies. Thus, leaving out the synthetical events in between we are interested in all permutations of such steps fulfilling the data dependencies to generate valid process variants. However, there are some specific events like "E8", "E9" or "E14" which restrict possible permutations in the sense that they play the role of a boundary functions which cannot be passed when moving up or down. Critical parts of the workflow are e.g. at functions "F10" where the contract is signed by the relationship manager and "F12" where the contract is approved by the credit officer. Here, the four-eye principle applies as a security requirement and demands that these functions have to be performed by two different persons. For this reason, the continuity processes have to ensure this requirement, too. It can be codified as a graph constraint as will be shown in Sect. 5.1.

In Fig. 4 a continuity snippet is presented for the function *rating customer* (F6). If the *rating application* that executes the business function becomes unavailable the process fragment that starts with the business event *credit worthiness computed* (E6) and ends with the business event *rating computed* (E7) can be replaced by the continuity snippet where the *customer officer* will execute the business function *rating customer* (F6(C)) as a back-up solution. This approach directly came up based on organizational requirements at Credit Suisse where knowledge about continuity scenarios is not necessarily provided by the same people that manage the standard business process and where knowledge is distributed among different persons. Our approach based on continuity snippets supports such a scenario because it can be codified in a declarative way by different people, collected and integrated as well as updated. The corresponding technical report (Brandt et al. 2010a) shows the generated graph grammar specifying the operational semantics consisting of 17 transformation rules as well as the complete set of 16 continuity snippets for this process model.

5 Generation of Business Continuity Processes

At Credit Suisse continuity plans are defined only for certain fixed scenarios at a macro level. Even so, there are lots of small failures in such a scenario. That leads to problems when more than one continuity scenario is going to come up at the same time, or when failures that are assigned to different scenarios mix up in a new scenario. In addition to that Credit Suisse's continuity plans are to a large extent not operational but need to be instantiated towards a concrete situation. That makes it impossible to check their effectiveness and efficiency. Because continuity processes based on a given continuity plan are created ad-hoc at the bank their effectiveness and efficiency cannot be evaluated either. So, Credit Suisse knows very little about the quality of its own continuity procedures, which drives insurance costs.

By looking at the human-centric business process model of the presented loan granting process we like to show how to address these problems in a methodological way. In contrast to Credit Suisse's approach which is top-down, we go bottom-up by defining continuity snippets for the presented business process. A second difference is that Credit Suisse's approach is static, which means it only addresses one very specific scenario, whereas ours is flexible and can address a wide range of different scenarios by applying continuity snippets to emerging failures in unforeseen ways by the people in the field. A third difference is that Credit Suisse's approach does not scale well, it always requires to execute the whole continuity plan, whereas our approach can dynamically focus on very specific parts of a scenario. It only needs to apply continuity snippets that are linked to concrete failures, not to whole scenarios. Therefore, we use the snippets as well as security rules like the one for the four-eye principle to generate the universe of business continuity processes given the universe of all functional valid process variants. Once this universe is available single continuity processes as well as the whole set can be checked regarding their effectiveness and efficiency. We will focus on the generation of valid process variants as well as possible continuity processes. The check for their effectiveness and efficiency is left for future work.

5.1 Analysis of Dependencies and Constraints

The first step for the generation of business continuity processes is the analysis of existing dependencies between the single steps of a process. Here, we focus on data-flow dependencies between functions. We therefore remove the dependencies caused by the syntactic events between the functions and only keep specific event-dependencies, which are marked as e.g. "E14" or which are used for a control structure as e.g. "E8" and "E9". This analysis allows for possible process paths that

show a different order of the steps and possibly enable the exchange of certain parts with continuity fragments that were not possible in the standard order. Furthermore, the process is kept flexible which improves the usability for the actors assuming that the execution of the process is supported by a workflow engine.

The analysis of the dependencies is performed by reconstructing a specific kind of graph grammar from the given WDEPC together with the continuity snippets as presented in (Brandt et al. 2009a, 2010a). This reconstruction can be performed automatically. The used underlying abstract syntax of WDEPC models can be automatically created during the modeling phase by e.g. using a visual editor that is generated by the Tiger environment (Biermann, Ermel, Lambers, Prange, and Taentzer 2010, Tiger Project Team 2009) based on a graph grammar specification. Furthermore, we can define the functional requirements as an additional graph rule, by stating which data elements have to be created in the process for a certain combination of events. In the presented example we have the requirement that either the customer is not accepted or the contract shall be approved and a closing date has to be set.
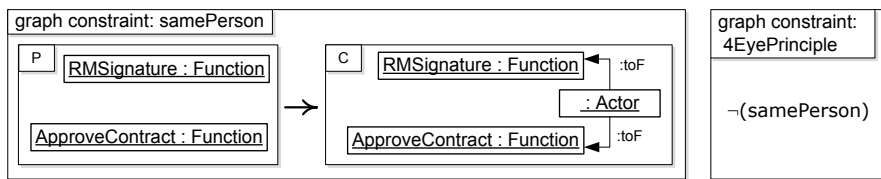

Fig. 5 Security Requirement: "the 4-eye principle"

Security constraints have to be respected by all possible process variants. As an important example we illustrate the handling of the four-eye principle specified by the constraint in Fig. 5. The functions "RM Signature (F10)" and "Approve Contract (F12)" have to be performed by different persons in order to ensure that core decisions are made by at least two persons. The graph constraint is given by a *negation* of the formal constraint "samePerson", which states the following: If the premise P is fulfilled, then also the conclusion graph C has to be found, i.e. in this case the two functions are executed by the same person as specified by the connecting edges. And indeed, the complete example with all continuity snippets can lead to a situation where the four-eye principle is not respected. There are several continuity snippets for these functions and in the case that both the relationship manager and the credit officer are temporarily unavailable, there is one combination of snippets where both actors are replaced by the credit advisor. The technique for the generation of continuity processes ensures the four-eye principle by checking the graph constraint on-the-fly. Moreover, nested graph constraints have been shown to be equivalent to first order logic expressions on graphs. Credit Suisse requested security requirements to be modeled in such a declarative way.

In the next step a Petri net is generated from the constructed grammar as presented in (Hermann, Corradini, Ehrig, and König 2010). Petri nets are a fundamental concept for specifying concurrent systems and provide efficient analysis techniques. The generated net purely specifies the dependencies between the process functions and is used to generate the universe of the business process variants and the possible continuity processes.

5.2 Generated Universe of Continuity Processes

Once the process model given by a WDEPC is analyzed as described above, the reachability graph of the derived dependency net can be generated, where functional as well as security requirements are respected in the way that steps that violate the security requirements are not performed and
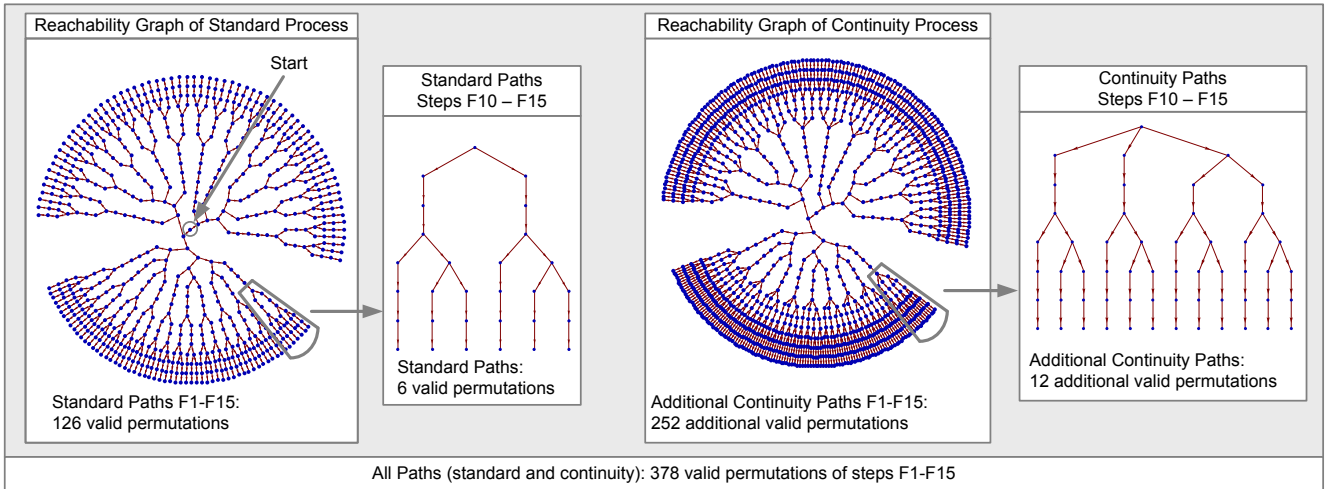
Fig. 6 Universe of Continuity Paths

paths that finally do not fulfill the functional requirements are filtered out. This way, the universe of valid process paths is generated as shown in Fig. 6. Graph techniques are well suited to check for *local* requirements and we show in Sect. 6 how a machine-centric business process model based on process algebra is generated out of the given WDEPC and checked for *global* requirements like data-flow and information-flow aspects.

The first graph in Fig. 6 shows the possible paths of the standard process in Fig. 3, where the middle node represents the starting point. Each arrow in the graph represents one step. At first, there are two choices for executing a function, namely, either the relationship manager records the customer identity or the customer demand. In any case the next step can be the retrieval of the other data or the storage of the already recorded data. This flexibility is not directly present in the WDEPC model, but by leaving out the syntactic events that fix the order of the steps, we derive this user-friendly flexibility.

The overall amount of possible sequences of the standard process is 126. All those sequences are *semantically equivalent* with respect to the given functional requirements. The second graph (from left) shows the last 6 functions including the loop at function "F14" leading to 7 steps for each path. Here again, there are two possibilities at the beginning: either the relationship manager signs the contract first or the customer. Usually the relationship manager will sign first, but there might also be few cases in which this is not the case because of time constraints of the customer. Finally, we derive 6 functional valid sequences for this part of the workflow.

The graphs on the right hand side of Fig. 6 show the additional continuity process paths that are possible using one or more continuity snippets for the functions "RM Signature (F10)" and "Approve Contract (F12)". The involved actors may be unavailable and the functions can be replaced using some of the continuity snippets given in (Brandt et al. 2010a). These functions have to respect the four-eye principle as discussed at the end of Sect. 5.1 before. Indeed, there is a combination of snippets in which both actors are replaced by the credit advisor. For this reason, the generation checks the security constraint shown in Fig. 5 on-the-fly and provides only the valid sequences. The amount of additional continuity paths is 252 and for the last 6 functions of the workflow part there are 12 additional paths.

There are certain advantages having all variants of the business process and its corresponding continuity processes generated. Firstly, in case of an emergency the Credit Suisse management can look at different options and make informed decisions which is not possible today. This leads to some sort of recommender service. Secondly, by having all continuity processes generated, continuity plans at Credit Suisse can be checked regarding their effectiveness and efficiency leading to
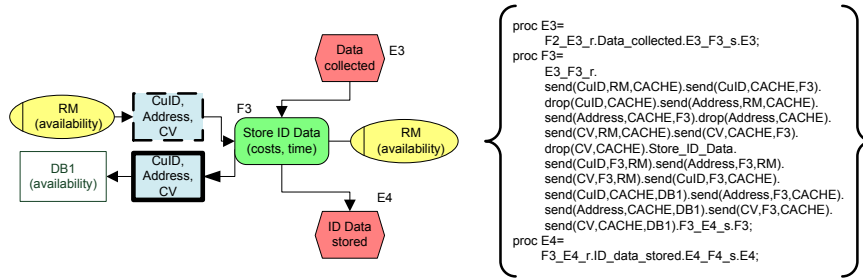
better optimized continuity plans. We can imagine to use Monte-Carlo simulations regarding the failures in a business process given its corresponding continuity snippets. Thirdly, given the different snippets and side-constraints the universe of process variants and corresponding continuity processes can be generated beforehand making it possible for a workflow engine to react without any delay towards upcoming failures. So, this approach is compatible with real-time requirements that are showing off in certain financial transactions at Credit Suisse. Fourthly, the business process model and its corresponding continuity snippets can be stored separately, which reduces the complexity of process models and supports ideally the bank's decentralized modeling workflows. It further makes the administration of enterprise models more intuitive.

6 Evaluation of Business Processes

In order to evaluate *global* properties of the human-centric business process model of the loan granting process shown in Fig. 3 we transform it into a machine-centric business process model formulated in *mCRL2* (Groote, Mathijssen, Reniers, Usenko, and van Weerdenburg 2008), which provides in this way a formal and denotational semantics for WDEPCs (see (Brandt et al. 2010a) for the full translation). The model transformation is based on triple graph transformation (Ehrig et al. 2009, Brandt, Hermann, and Engel 2009b) – a special kind of algebraic graph transformation. This allows to declaratively specify by triple rules how certain structures in the abstract syntax of the human-centric business process model are mapped towards structures in the abstract syntax of the machine-centric business process model.

6.1 Machine-Centric Business Process Model

A machine-centric business process model captures very specific qualities of the given business process in order to enable *global* checks by keeping its overall state space small. Therefore, there may be more than one *mCRL2* model needed depending on the purpose in mind as it can be seen in (Boehmer, Brandt, and Groote 2009) where an *mCRL2* model for a business process is presented to simulate timing behavior. Figure 7 shows an WDEPC fragment and its corresponding *mCRL2* fragment, derived by the model transformation for data-flow checks between agents.


```
proc E3=
    F2_E3_r.Data_collected.E3_F3_s.E3;
proc F3=
    E3_F3_r.
    send(CuID,RM,CACHE).send(CuID,CACHE,F3).
    drop(CuID,CACHE).send(Address,RM,CACHE).
    send(Address,CACHE,F3).drop(Address,CACHE).
    send(CV,RM,CACHE).send(CV,CACHE,F3).
    drop(CV,CACHE).Store_ID_Data.
    send(CuID,F3,RM).send(Address,F3,RM).
    send(CV,F3,RM).send(CuID,F3,CACHE).
    send(CuID,CACHE,DB1).send(Address,F3,CACHE).
    send(Address,CACHE,DB1).send(CV,F3,CACHE).
    send(CV,CACHE,DB1).F3_E4_s.F3;
proc E4=
    F3_E4_r.ID_data_stored.E4_F4_s.E4;
```

Fig. 7 Model transformation from a WDEPC-fragment to an *mCRL2* code snippet

Here, the business events "E3", "E4" and the function "F3" are mapped towards the process variables "E3", "F4" and "E4". The data-flow between the process variables is realized by *send* actions in the *mCRL2* code visible in Fig. 7. *Drop* actions show that data-items are forgotten by the cache of the workflow engine.

6.2 Model Transformation using Triple Rules and derived Forward Rules

Because the people in the field do not want to touch the *mCRL2* specification directly, we like to show how it can be derived by model transformation without human intervention. For this purpose
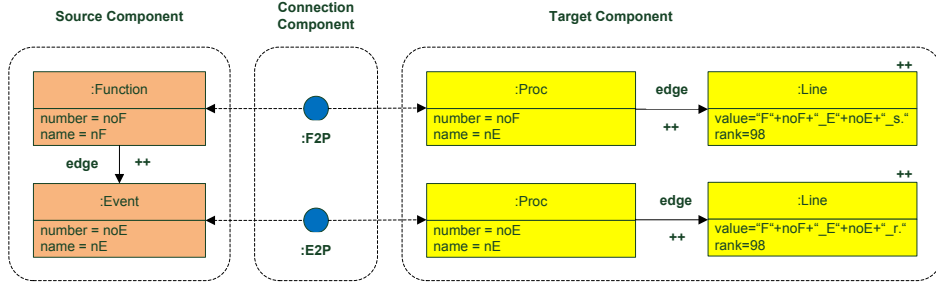
Fig. 8 Mapping the edge between a function and an event to *mCRL2* code by a triple graph rule

we present one triple graph rule in Fig. 8 that demonstrates how a directed edge between a business function and a business event in the WDEPC model is translated into two lines of *mCRL2* code. This is done by showing how a source and a target model can be constructed simultaneously by associating construction steps at both sides by the help of connection nodes. Therefore, a triple graph rule consists of a source, a target and a connection component. Here, it is presented in its compact notation, which means that elements annotated with "++" are added after matching the elements without annotation to the already existing graphs of the abstract syntax of models at both sides. So, after matching the function and event nodes and their associated nodes for process variables, the directed edge is added in the source model and two lines of *mCRL2* code are added in the target model as well as the edges that link these lines to the nodes of the corresponding process variables. In order to keep the rules clear and simple for this presentation this model transformation produces the abstract syntax of an intermediate model that can directly be translated into *mCRL2*. The boxes represent certain types and their corresponding attributes. So, given the name and number of a function and of an event as well as the number and name of the matched process variables, the two new *mCRL2* lines can be constructed and stored in the corresponding value attribute. Their pre-defined rank specifies that the set of generated lines by all triple graph rules will finally be placed in the right order.

6.3 Data-Flow and Information-Flow Evaluations

Once, the application of forward rules derived from the given triple rules to the abstract syntax of the human-centric process model results in the corresponding machine-centric process model, we can model-check it in order to determine whether the credit advisor is able to know the data containing the address of the customer (cf. (Groote et al. 2008)). This is checked by the help of the modal logic formula in Fig. 9, which states that the credit advisor cannot know the address, and this condition is evaluated to true when applied to the *mCRL2* specification coming out of the model transformation for data-flow evaluations. More precisely, the formula maintains a set of knowledge `knowledge_set` consisting of pairs of which datastring is known by who. E.g. `pair(Address, CA) in knowledge_set` expresses that the address string is known by the credit advisor CA. When data is generated (`generate`) or sent (`send`) it becomes known to more parties,

```
1  nu X(knowledge_set:Set(LocatedKnowledge)={}).!(val(pair(Address,CA) in knowledge_set)) &&
2  (forall d:DataString, s:Store.[generate(d,s)] X(AddInformation(d,s,knowledge_set))) &&
3  (forall d:DataString, s:Store.[drop(d,s)] X(knowledge_set)) &&
4  (forall d:DataString, s1,s2:Store.[send(d,s1,s2)] X(AddInformation(d,s2,knowledge_set))) &&
5   [(forall d:DataString, s:Store.!generate(d,s)) &&
6  (forall d:DataString, s:Store.!drop(d,s)) &&
7  (forall d:DataString, s1,s2:Store.!send(d,s1,s2))] X(knowledge_set)
```

Fig. 9 Specification in *mCRL2* for "credit advisor cannot know the data with the customer address"

and hence, it is added to the knowledge set. If anything else happens to the data, such as dropping the data from some store, the knowledge set is not changed.

Likewise, the global information flow can be checked by the help of an *mCRL2* model. Information is defined as an abstract object associated with given data. As such, information can be passed to other data items when data sets are processed together while a business functions executes. Therefore, information-flow analysis requires a specified data-flow. Our model assumes that a certain percentage of information from one data item can be passed to another one. So, that, at the end, we can check, how much information will be available to different agents at different steps in the process. The model transformation and *mCRL2* specification for *global* data- *and* information-flow checks as well as a corresponding modal logic formula are given in (Brandt et al. 2010a).

6.4 State Space of the mCRL2 model

The *mCRL2* model consists of 41 parallel processes representing tasks and events. The largest process ("F1", Get Customer Identity) has 21 actions consisting of passing information around and waiting for and triggering events. Atypically, there is not very much parallel behaviour as the processes get triggered, carry out their tasks and trigger the next process. This means that the state space is rather concise, namely only 211 states. The time to generate the states space is neglectable. The only non trivial aspect is to eliminate the universal quantifiers in the formula in Fig. 9. After elimination the verification of this and other modal formulae was almost instantaneous.

Fig. 10 A reduced state space of the workflow showing 35 of 211 states

A state space with 211 states is already too large to visualise explicitly, but if one makes abstractions or projections of the state space, one can get insight in almost any conceivable behavioural aspect of it. We provide two illustrations. If we hide all actions, except task and event names and reduce the resulting state space modulo branching bisimulation, we obtain the state space depicted in Fig. 10, which has 35 states. The initial state is at the left-lower corner. Transitions labelled with `tau` are hidden actions that determine a choice in behaviour and can therefore not be removed.

Sometimes it is more desired to get insight per component. For this the tool *Diagraphica* (Pretorius and van Wijk 2006) can be used. In Fig. 11 the workflow is projected onto tasks "F1" and "F2". The states of the whole graph are projected into the 21 states of "F1". They are drawn

Fig. 11 State space projected onto tasks "F1" and "F2" using *Diagraphica*, direction of arches: clock wise

at the top. Within that it is shown how the states are projected into the different states of task "F2". Following the arches in a clock wise fashion shows how one can move among the projected states. These arches generally represent large numbers of transitions. This diagram stresses the sequential nature of this system. Only if "F1" is in its leftmost state, "F2" can move from state to state.

7 Evaluation of the Proposed Approach

This section will reflect shortly to what degree the results presented in this paper have been proven useful in the particular case of Credit Suisse. Concretely, it was required to use EPCs or something very similar to EPCs to be able to work with already existing business process models. For organizational and usability reasons it was requested to provide a declarative way to specify continuity snippets that are simplified as much as possible because they can be combined by the help of the formal graph technique to more complex ones automatically. Further, it was requested to show if and how the universe of business continuity processes can be generated in order to prepare for business continuity simulations and recommendation services later on. Finally, the evaluation of single business continuity processes was asked for regarding its local and global process properties.

Based on a simplified real-world loan granting process we were able to show how these requests can be addressed. Our specified and (partly) implemented proof of concept illustrates how formal tools based on different but complementary formal techniques of algebraic graph transformation and process algebra encompassing modal logic can be used in a combined way to automatically generate business process variants, derive possible continuity processes and result into sound evaluations. This setting contrasts conventional software engineering solutions that do not lead to formal guarantees requested by Credit Suisse. In addition to the above, and because these formal tools are by nature generic, they provide support for open and evolving families of domain languages as requested in (Brandt et al. 2009c). Further, this approach is fully compatible with the idea of reusing the functionality of an upcoming graph transformation engine that is currently under construction (Braatz and Brandt 2008, 2010) and which will enable the analysis of much more complex business processes in the near future that therefore have not been presented in the scope of this paper yet. Our result enables Credit Suisse to ensure that the used processes out of the generated universe of business continuity processes respect the given *local* (4-eye principle) and *global* (data- and information flow) security requirements. The first case was ensured by graph constraints supported by the formal technique of algebraic graph transformation. The second case was checked by model checking techniques of the *mCRL2* tool suite. As requested by Credit Su-

isse, the presented approach build up on top of formal tools to benefit from formal guarantees that help to ensure security requirements.

8 Related Work

Finally, we like to give some selected pointers into the area of business continuity, enterprise modeling, business process modeling and business process semantics to show how our work is related to the work of others and the current industrial practices. We do not claim to link to the whole variety of available approaches, but like to show the context of our approach discussed from the point of view of Credit Suisse.

8.1 Business Continuity

The main purpose of business continuity at Credit Suisse is to protect the organization from all kinds of failures and disasters by keeping the organization, its critical business processes, operational (Nemzow 1997, Zalewski et al. 2008). Non-integrated continuity plans did not succeed here (IBM 2006). An overview of known approaches can be found in (Quirchmayr 2004).The main problem is a lack of preparation as well as unrealistic assumptions (Landry and Koger 2006b). Today's continuity plans are often just textual documents that cannot be verified as e.g. by simulation (Tjoa, Jakoubi, and Quirchmayr 2008) or a resource and data driven analysis (Nigam and Caswell 2003). However, according to Credit Suisse continuity plans need to be checked for security, risk and compliance requirements(Quirchmayr 2004, Cha, Juo, Liu, and Chen 2008). One available solution able to address such questions is the ADEPT system (Reichert and Dadam 1998). In detail, it guarantees the structural correctness of a process model while applying dynamic changes and in (Rinderle-Ma, Reichert, and Weber 2008) a trace based semantics is given for general change patterns for which the effect must be precisely defined. Moreover, the ADEPT system provides an interface for end users to perform general modifications of the process graph (Dadam and Reichert 2009) and checks data dependencies before performing a transformation. A context-aware selection of options including preconfigured and checked process variants are available for the Provop approach presented in (Hallerbach, Bauer, and Reichert 2009), where users specify the dependencies between the available options by constraints. Further checks for basic semantic constraints concerning dependencies and mutual exclusions are performed in the SeaFlows project (Ly, Rinderle, and Dadam 2008). However, our presented approach puts its emphasis on orthogonal issues. It is built up on top of a holistic framework for organizational enterprise models and focuses on the integration of formal methods supporting modeling activities with formal methods supporting analysis activities in a way that is – in principle – usable by the people in the field. The proposed formal methods show good potential to be fully implemented or are already implemented as tools.

In order to keep the complexity of the generation of continuity processes in bounds that are suitable for the application domain and to keep the specification efforts manageable we do not consider general process transformations. Our generation creates all possible reordering of the given standard process in combination with possible replacements using the continuity process snippets, where each snippet handles one particular unavailability of a resource or an actor. General transformations can be combined with our approach in an orthogonal way. The generated process continuations for each combination of failures are ensured to be equivalent with respect to the functional requirements based on the formal notion of permutation equivalence and they conform to the given *local* and *global* side constraints. All dependency constraints are ensured on-the-fly, which improves the efficiency.

Currently available approaches are not able to satisfy the requirement of Credit Suisse to use fully automated formal methods that are used in a declarative and transparent way compatible with organizational side-constraints like decentralization. Moreover, there is a special need for simulation and formal verification in order to ensure conformance and to provide validated recommender services. In the Credit Suisse scenario a sufficiently complete set of continuity processes has to be generated in advance, without user interaction and based on simple continuity snippets to enable automatic optimizations and case based decisions with the knowledge that the given side-constraints are ensured.

8.2 Enterprise Modeling

The main purpose of enterprise modeling at Credit Suisse is to support the development and maintenance of enterprise wide IT systems (Lankhorst 2005). It is therefore implementation-driven. Here, we claim that organizational models (Sousa, Van Aken, and Rentes 2001) are the primary object of interest that can be used for implementations or automation later on as some sort of by-product. That contradicts the practice at the bank. We assume to work with small, simple and decentralized models (Pereira and Sousa 2004), which can be automatically integrated, and that do not necessarily match with the wide variety (Giaglis 2001) of paradigms, frameworks and architectures (Sousa et al. 2001, Lillehagen and Krogstie 2008) that are partly build on big, complex and centralized organizational models that are integrated right from their beginning due to a current lack of knowledge in modeling techniques (Camarinha-Matos, Afsarmanesh, and Ollus 2005). So, here, we like to look at enterprise modeling as a means to properly document the business model of Credit Suisse, which runs on top of an IT infrastructure that can be owned by the bank but need not to be managed by it. This implies the need for an open family of domain languages. For this purpose, we use the formal approach of graph transformation for the abstract syntax of the models and triple graph grammars for model based interoperability (Brandt et al. 2009b).

8.3 Business Process Modeling

The main purpose of business process modeling at Credit Suisse is automation and evaluation of compliance (Brandt, Otten, Kreitz, and Bibel. 2010b). It goes along with business object modeling (Caetano, Silva, and Tribolet 2005). The main problems among others are over-engineering and a lack of specialization of the resulting business process models (Rosemann, Recker, Indulska, and Green 2006) which is partly caused by a chaos of modeling techniques (Giaglis 2001) that are applied without any deep theoretical understanding (Camarinha-Matos et al. 2005). In detail, there is no clear separation of concerns between business and IT processes as well as their sound integration (Giaglis, Paul, and Serrano 1999). Adaptable formal foundations for any kind of evaluation are missing (Zhang and Wang 2006, Wong and Gibbons 2007, Kawakami, Akinaga, Suto, and Katai 2003) to enable model-checking (Dai, Bai, and Zhao 2007). Model reuse by the help of reference models and corresponding process configurations as well as an assisted construction process are not supported (Thomas, Horiuchi, and Tanaka 2006, Gottschalk, Wagemakers, Jansen-Vullers, van der Aalst, and Rosa 2009, van der Aalst et al. 2008, Bhattacharya, Gerede, Hull, Liu, and Su 2007). The flexible integration of evolving domain models of different types like processes and rules as well as their evolving domain languages is not realized (Börger and Thalheim 2008, Zaha, Barros, Dumas, and ter Hofstede 2006). Our approach shows good potential to address lots of these issues by formal techniques based on algebraic graph transformation in an automated or semi-automated as well as a transparent way. The chosen modeling language of extended EPCs

was driven by Credit Suisse based on the knowledges and experiences of the domain experts. The presented formal techniques based on the abstract syntax, however, can be applied to other DSLs like e.g. BPMN (Object Management Group 2010) as well. In the context of this study, the explicit modeling of actors within EPCs is a key feature important for process models at Credit Suisse.

8.4 Semantics and Analysis of Business Processes

The semantics and analysis of business processes is often discussed using petri-nets, colored petri-nets, the pi-calculus, CSP or Z (van der Aalst 1999, Gottschalk, van der Aalst, Jansen-Vullers, and Verbeek 2008, Puhlmann 2006, Nestmann and Puhlmann 2008, Wong and Gibbons 2008). However, there are several other semantics available. Up to now, tool overviews for business processes do not mention the process algebraic tool suites because they are generally not used for evaluations of functional and non-functional properties of business processes. See for instance the overview of Jansen-Vullers and Netjes (Jansen-Vullers and Netjes 2006). However, we believe that classical data enlarged process algebras such as *mCRL2* are certainly as suitable for modelling and analysis of business processes.

9 Conclusion: Summary and Future Work

Our approach presents how fully automated formal methods can be transparently used by people at Credit Suisse to model continuity plans that consists of business process models and their corresponding continuity snippets. In detail, we claim that business processes should be administrated as evolving domain models by the help of their corresponding domain languages. Implemented formal generation techniques create the whole set of possible process variants as well as all possible continuity processes that respect given side-constraints. By transforming such human-centric process models into machine-centric process models fully implemented formal methods like *mCRL2* can be used to evaluate the data- and information-flow of a process fully transparent to the people in the field as it was requested by the management of the bank. By using algebraic graph transformation as a formal technique for model transformation, integration and synchronization as well as construction we are able to work with small declarative rules, which support the encoding of decentralized knowledge about enterprise models and the underlying modeling process in a way that ideally fits organizational requirements of the bank. From the specific point of view of business continuity the integration of our approach into an overall framework for enterprise modeling enables large scale reuse of organizational models as well as consistency checks between models. Future work will encompass the development of small domain languages for services, processes and rules that can smoothly be integrated by techniques of algebraic graph transformation and integrated with purposeful machine-centric models in a transparent way.

References

Y. Asnar and P. Giorgini. Analyzing business continuity through a multi-layers model. In *BPM '08: Proc. of the Int. Conf. on Business Process Management*, pages 212–227. Springer, 2008.

Basel Committee on Banking Supervision. *Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework - Comprehensive Version*, 2006. http://www.bis.org/publ/bcbsca.htm.

K. Bhattacharya, C. E. Gerede, R. Hull, R. Liu, and J. Su. Towards formal analysis of artifact-centric business process models. In G. Alonso, P. Dadam, and M. Rosemann, editors, *Proc. Int. Conf. on Business Process Management (BPM'07)*, volume 4714 of *LNCS*, pages 288–304. Springer, 2007.

E. Biermann, C. Ermel, L. Lambers, U. Prange, and G. Taentzer. Introduction to AGG and EMF Tiger by Modeling a Conference Scheduling System. *Software Tools for Technology Transfer*, 2010. To appear.

W. Boehmer. Survivability and business continuity management system according to bs 25999. In *SECURWARE '09: Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pages 142–147. IEEE Computer Society, 2009.

W. Boehmer, C. Brandt, and J. F. Groote. Evaluation of a business continuity plan using process algebra and modal logic. In *IEEE Toronto International Conference on Science and Technology for Humanity (IEEE-TIC-STH)*, SIASP 2, pages 147–152. IEEE, IEEE Computer Society, 2009.

E. Börger and B. Thalheim. A method for verifiable and validatable business process modeling. *Advances in Software Engineering*, pages 59–115, 2008.

B. Braatz and C. Brandt. Graph transformations for the resource description framework. *ECEASST*, 10, 2008.

B. Braatz and C. Brandt. How to modify on the semantic web? - a web application architecture for algebraic graph transformations on rdf. In *2nd International Workshop on Semantic Inforamtion Management – SWIM at ICWE 2010*, 2010.

C. Brandt, F. Hermann, and T. Engel. Modeling and reconfiguration of critical business processes for the purpose of a business continuity management respecting security, risk and compliance requirements at credit suisse using algebraic graph transformation. In *Enterprise Distributed Object Computing Conference Workshops, Proc. Int. Workshop on Dynamic and Declarative Business Processes (DDBP'09)*, pages 64 –71. IEEE Xplore Digital Library, 2009a.

C. Brandt, F. Hermann, and T. Engel. Security and Consistency of IT and Business Models at Credit Suisse realized by Graph Constraints, Transformation and Integration using Algebraic Graph Theory. In *Enterprise, Business-Process and Information Systems Modeling, Proc. Int. Conf. on Exploring Modeling Methods in Systems Analysis and Design 2009 (EMMSAD'09)*, volume 29 of *LNBIP*, pages 339–352, Heidelberg, 2009b. Springer Verlag.

C. Brandt, F. Hermann, and T. Engel. Security and consistency of it and business models at credit suisse realized by graph constraints, transformation and integration using algebraic graph theory. In T. A. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, and R. Ukor, editors, *BMMDS/EMMSAD*, volume 29 of *Lecture Notes in Business Information Processing*, pages 339–352. Springer, 2009c.

C. Brandt, F. Hermann, and J. F. Groote. Modeling and Reconfiguration of critical Business Processes for the purpose of a Business Continuity Management respecting Security, Risk and Compliance requirements at Credit Suisse using Algebraic Graph Transformation: Long Version. Technical Report 2010-11, Technische Universität Berlin,Fakultät IV, 2010a. To appear, online: http://www.tfs.cs.tu-berlin.de/publikationen/Papers10/BHE10.pdf.

C. Brandt, J. Otten, C. Kreitz, and W. Bibel. Specifying and verifying organizational security properties in first-order logic. In S. Siegler and N. Wasser, editors, *Verification, Induction, Termination Analysis*, LNCS. Springer, 2010b. To appear.

A. Caetano, A. R. Silva, and J. Tribolet. Using roles and business objects to model and understand business processes. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 1308–1313. ACM, 2005.

L. M. Camarinha-Matos, H. Afsarmanesh, and M. Ollus, editors. *Virtual Organizations.* Springer, 2005.

W. Cassel. Assessing business continuity requirements. In *Power Engineering Society General Meeting*, volume 1, pages 822–823. IEEE Computer Society, 2005.

S.-C. Cha, P.-W. Juo, L.-T. Liu, and W.-N. Chen. Riskpatrol: A risk management system considering the integration risk management with business continuity processes. In *Intelligence and Security Informatics*, pages 110 – 115. IEEE, 2008.

A. Corradini, F. Hermann, and P. Sobociński. Subobject Transformation Systems. *Applied Categorical Structures*, 16(3): 389–419, 2008.

P. Dadam and M. Reichert. The adept project: a decade of research and development for robust and flexible process support. *Computer Science - R&D*, 23(2):81–97, 2009.

G. Dai, X. Bai, and C. Zhao. A framework for model checking web service compositions based on bpel4ws. In S. C. Cheung, Y. Li, K.-M. Chao, M. Younas, and J.-Y. Chung, editors, *ICEBE*, pages 165–172. IEEE Computer Society, 2007.

H. Ehrig, C. Ermel, F. Hermann, and U. Prange. On-the-Fly Construction, Correctness and Completeness of Model Transformationsbased on Triple Graph Grammars. In A. Schürr and B. Selic, editors, *ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09)*, volume 5795 of *LNCS*, pages 241–255. Springer, 2009.

G. M. Giaglis. A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques. *International Journal of Flexible Manufacturing Systems*, 13(2):209–228, 2001.

G. M. Giaglis, R. J. Paul, and A. Serrano. Reconciliation of business and systems modelling via discrete event simulation. In *WSC '99: Proceedings of the 31st conference on Winter simulation*, pages 1403–1409. ACM, 1999.

F. Gottschalk, W. M. P. van der Aalst, M. H. Jansen-Vullers, and H. M. W. Verbeek. Protos2cpn: using colored petri nets for configuring and testing business processes. *STTT*, 10(1):95–110, 2008.

F. Gottschalk, T. A. C. Wagemakers, M. H. Jansen-Vullers, W. M. P. van der Aalst, and M. L. Rosa. Configurable process models: Experiences from a municipality case study. In P. van Eck, J. Gordijn, and R. Wieringa, editors, *CAiSE*, volume 5565 of *LNCS*, pages 486–500. Springer, 2009.

J. F. Groote, A. Mathijssen, M. A. Reniers, Y. S. Usenko, and M. van Weerdenburg. *Analysis of Distributed Systems with mCRL2*, chapter 4, pages 99–128 (www.mcrl2.org). 2008.

A. Hallerbach, T. Bauer, and M. Reichert. Guaranteeing soundness of configurable process variants in provop. In B. Hofreiter and H. Werthner, editors, *Proc. IEEE Conf. on Commerce and Enterprise Computing, (CEC'09)*, pages 98–105. IEEE Computer Society, 2009.

R. W. Helms, S. Van Oorschot, J. Herweijer, and M. Plas. An integral it continuity framework for undisrupted business operations. In *Proc. Int. Conf. on Availability, Reliability and Security (ARES'06)*, pages 240–244. IEEE Computer Society, 2006.

F. Hermann, A. Corradini, H. Ehrig, and B. König. Efficient Analysis of Permutation Equivalence of Graph Derivations Based on Petri Nets. In J. Küster and E. Tuosto, editors, *Proc. Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT'10)*. EC-EASST, 2010. To appear.

IBM. Panic slowley. integrated disaster response and bulit-in business continuity, 2006. http://www.ibm.com/itsolutions/uk/governance/businesscontinuity/.

M. Jansen-Vullers and M. Netjes. Business process simulation - a tool survey. In K. Jensen (ed.). Seventh Workshop and Tutorial on the Practical Use of Coloured Petri Nets and the CPN Tools (CPN' 06), Volume 579 of DAIMI, pp. 77-96. University of Arhus, Denmark, 2006.

H. Kawakami, R. Akinaga, H. Suto, and O. Katai. Translating novelty of business model into terms of modal logics. In T. D. Gedeon and L. C. C. Fung, editors, *Australian Conference on Artificial Intelligence*, volume 2903 of *LNCS*, pages 821–832. Springer, 2003.

R. J. Kepenach. Business continuity plan design. In *ICIMP '07: Proceedings of the Second International Conference on Internet Monitoring and Protection*, page 27. IEEE Computer Society, 2007.

W. Lam. Ensuring business continuity. *IT Professional*, 4:19–25, 2002.

B. J. L. Landry and M. S. Koger. Dispelling 10 common disaster recovery myths: Lessons learned from hurricane katrina and other disasters. *J. Educ. Resour. Comput.*, 6(4):6, 2006a.

B. J. L. Landry and M. S. Koger. Dispelling 10 common disaster recovery myths: Lessons learned from hurricane katrina and other disasters. *J. Educ. Resour. Comput.*, 6(4):6, 2006b.

M. Lankhorst. *Enterprise Architecture at Work. Modelling, Communication and Analysis*. Springer, 2005.

F. Lillehagen and J. Krogstie. State of the Art of Enterprise Modeling. In *Active Knowledge Modeling of Enterprises*, pages 91–127. Springer, 2008.

L. T. Ly, S. Rinderle, and P. Dadam. Integration and verification of semantic constraints in adaptive process management systems. *Data Knowl. Eng.*, 64(1):3–23, 2008.

M. Nemzow. Business continuity planning. *Int. J. Netw. Manag.*, 7(3):127–136, 1997.

U. Nestmann and F. Puhlmann. *Business Process Secification and Analysis*, chapter 5. Chapman & Hall/CRC (ed) Alexander, Michael and Gardner, William, 2008.

A. Nigam and N. S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3), 2003.

Object Management Group. *Business Process Model and Notation (BPMN)*, 2010. formal/10-06-05, `http://www.omg.org/spec/BPMN/2.0`.

C. M. Pereira and P. Sousa. A method to define an Enterprise Architecture using the Zachman Framework. In *Proc. ACM symposium on Applied Computing (SAC'04)*, pages 1366–1371, New York, NY, USA, 2004. ACM.

A. Pretorius and J. van Wijk. Visual analysis of multivariate state transition graphs. In *IEEE Transactions on Visualization and Computer Graphics vol. 12, no. 5*, pages 685–692, 2006.

F. Puhlmann. Why do we actually need the pi-calculus for business process management? In H. C. M. Witold Abramowicz, editor, *9th International Conference on Business Information Systems (BIS 2006)*, pages 77–89. Department of Information Systems, Poznań University of Economics, 2006.

G. Quirchmayr. Survivability and business continuity management. In *Proc. of the 2nd WS on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation (ACSW Frontiers '04)*, pages 3–6. Australian Computer Society, Inc., 2004.

M. Reichert and P. Dadam. ADEPT flex -supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems*, 10(2):93–129, 1998.

S. Rinderle-Ma, M. Reichert, and B. Weber. On the formal semantics of change patterns in process-aware information systems. In Q. Li, S. Spaccapietra, E. S. K. Yu, and A. Olivé, editors, *Proc. Int. Conf. on Conceptual Modeling (ER'08)*, volume 5231 of *LNCS*, pages 279–293. Springer, 2008.

M. Rosemann, J. Recker, M. Indulska, and P. Green. A study of the evolution of the representational capabilities of process modeling grammars. In E. Dubois and K. Pohl, editors, *CAiSE*, volume 4001 of *LNCS*, pages 447–461. Springer, 2006.

A.-W. Scheer, editor. *ARIS-Modellierungs-Methoden, Metamodelle, Anwendungen*. Springer, 2001.

A. Schürr. Specification of Graph Translators with Triple Graph Grammars. In G. Tinhofer, editor, *Proc. Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG'94)*, volume 903 of *LNCS*, pages 151–163. Springer, 1994.

G. Sousa, E. Van Aken, and A. Rentes. Using enterprise modeling to facilitate knowledge management in organizational transformation efforts. *Portland International Conference on Management of Engineering and Technology. IEEE.*, 1:62, 2001.

C. Stych and K. Zeppenfeld. ITIL, 2008.

O. Thomas, M. Horiuchi, and M. Tanaka. Towards a reference model management system for business engineering. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 1524–1531. ACM, 2006.

Tiger Project Team. EMF Tiger, 2009. `http://tfs.cs.tu-berlin.de/emftrans`.

S. Tjoa, S. Jakoubi, and G. Quirchmayr. Enhancing business impact analysis and risk assessment applying a risk-aware business process modeling and simulation methodology. In *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*, pages 179–186, Washington, DC, USA, 2008. IEEE Computer Society.

United States Code. Sarbanes-oxley act of 2002, pl 107-204, 116 stat 745. Codified in Sections 11, 15, 18, 28, and 29 USC, 2002.

W. M. P. van der Aalst. Formalization and verification of event-driven process chains. *Information and Software Technology*, 41(10):639–650, 1999.

W. M. P. van der Aalst, M. Dumas, F. Gottschalk, A. H. M. ter Hofstede, M. L. Rosa, and J. Mendling. Correctness-preserving configuration of business process models. In J. L. Fiadeiro and P. Inverardi, editors, *FASE*, volume 4961 of *LNCS*, pages 46–61. Springer, 2008.

S. Y.-H. C. Wan. Adoption of business continuity planning processes in it service management. In *3rd IEEE/IFIP International Workshop on Business-driven IT Management*, pages 21–30, Salvador, 2008. IEEE Computer Society.

P. Y. H. Wong and J. Gibbons. A process-algebraic approach to workflow specification and refinement. In M. Lumpe and W. Vanderperren, editors, *Software Composition*, volume 4829 of *LNCS*, pages 51–65. Springer, 2007.

P. Y. H. Wong and J. Gibbons. A process semantics for bpmn. In S. Liu, T. S. E. Maibaum, and K. Araki, editors, *ICFEM*, volume 5256 of *LNCS*, pages 355–374. Springer, 2008.

J. M. Zaha, A. P. Barros, M. Dumas, and A. H. M. ter Hofstede. Let's dance: A language for service behavior modeling. In R. Meersman and Z. Tari, editors, *OTM Conferences (1)*, volume 4275 of *LNCS*, pages 145–162. Springer, 2006.

A. Zalewski, P. Sztandera, M. Ludzia, and M. Zalewski. Modeling and analyzing disaster recovery plans as business processes. In *SAFECOMP '08: Proc. Int. Conf. on Computer Safety, Reliability, and Security*, volume 5219 of *LNCS*, pages 113–125. Springer, 2008.

E. Zambon, D. Bolzoni, S. Etalle, and M. Salvato. A model supporting business continuity auditing and planning in information systems. In *ICIMP '07: Proceedings of the Second International Conference on Internet Monitoring and Protection*, page 33. IEEE Computer Society, 2007.

J. Zhang and H. Wang. A pi-calculus-based business process formal design method. In W. Shen, J. Luo, Z. Lin, J.-P. A. Barthès, and Q. Hao, editors, *CSCWD (Selected Papers)*, volume 4402 of *LNCS*, pages 347–356. Springer, 2006.